A Neural Network Approach for Pre-classification in Musical Chords Recognition

Thierry Gagnon[†], Steeve Larouche and Roch Lefebvre[†]
[†]Speech and Audio Processing Lab
University of Sherbrooke, Electrical Engineering Department
2500, Université Boulevard, Sherbrooke, Quebec, Canada
Thierry.Gagnon@USherbrooke.ca, slarouche@hermes.usherb.ca,
Roch.Lefebvre@USherbrooke.ca

Abstract-**Automatic music transcription is a complex task involving signal processing, pattern-matching, signal classification, and the integration of musical knowledge. Existing systems use both Bottom-Up and Top-Down approaches. Their performance is determined by the parameters extraction algorithms and the number of different patterns to discriminate. We propose a neural network based pre-classification approach to allow a focused search in the chords recognition stage. The specific case of the 6-string standard guitar is considered. The pre-classification algorithm outputs the number of strings in a chord and the estimated hand position on the guitar neck.**

I. Introduction

Automatic music transcription is a difficult task involving signal processing, pattern matching, signal classification and the integration of musical knowledge. There have been significant advances over the past decades, with an emphasis on accurate time segmentation [3,5,7], multipitch analysis [1,5,9,11] and signal separation [4,6].

Recently, systems based on *a priori* musical knowledge have produced interesting results in the case of a single instrument [7]. However, state-of-the-art algorithms still have shortcomings. In particular, even in the case of a single instrument, chords with more than four simultaneous notes are hard to precisely identify. Also, the more robust approaches use a closed-loop design which makes them difficult to implement in an interactive, real-time application.

The paper is organized as follows. Section II briefly explains the chords composition. Section III presents a general transcription system ; while section IV describes the proposed pre-classification approach based on parallel neural networks. Results for synthetic musical chords are presented in section V. Finally, section VI gives some conclusions, and possible directions for future work.

II. Chords Composition

In the case of a string instrument, each note (or vibrating string) is formed by summing harmonics, convolved by a spectral envelope (figure 1). The spectral envelope is related to the instrument tonality, or its "color". For the 6-string 22-fret guitar, the fundamental frequency of an individual note ranges from 82.41 to 1244.51 Hertz. This corresponds to all possible finger positions (or notes) on the fingerboard. Figure 1 shows that the prominent spectral components are beneath 2 kHz. The high-frequency components are considerably attenuated by the spectral envelope.

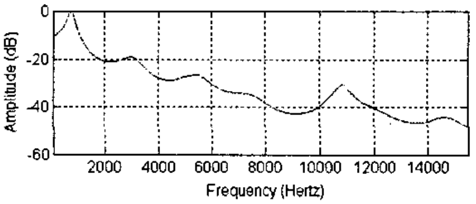

Fig. 1. Spectral Envelope Example for the 6-String, 22-Fret Guitar

Harmonic Overlap

In a chord, the harmonics from different notes can overlap each other. For example, the first four harmonics of the *A* note are at frequencies 110, 220, 330 and 440 Hertz. Similarly, the first four harmonics of the *E* note are at frequencies 82.4, 164.8, 247.2 and 329.6 Hertz. Obviously, the third harmonic of the *A* note will overlap the fourth harmonic of the *E* note. The frequency resolution necessary to resolve these two harmonics would require long time support [1,11].

Symbolic Representation

Figure 2 shows the symbolic representation of a guitar chord : the fingering chart, or "tablature" (from French).

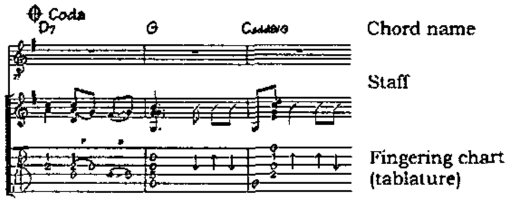

Fig. 2. Fingering chart example

To generate the synthetic chords for training our system, we use this representation (figure 2), especially because of its numerical notation. It is easy to generate chords from real or synthetic signals databases by choosing one fingering chart, and transpose it to all the other hand positions. The fingering chart is used to randomly choose the chord for training or simulation purposes.

0-7803-8104-1/03/$17.00 ©2003 IEEE

III. General Transcription System

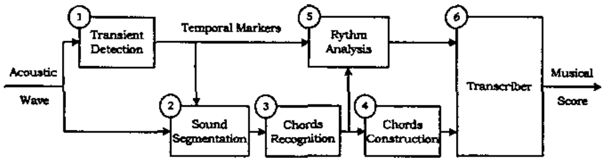

Fig. 3. Global Block Diagram of the Transcription System

Figure 3 shows a general block diagram of an automatic music transcription system. The input audio samples are first segmented in successive chords or notes (blocks 1 and 2). Each individual segment forms the entry of the chords recognition module (block 3). In our case, the Recognition Module will be further decomposed in a pre-classification module and focused search module as described below. Finally, logic and musical knowledge is applied to obtain the final musical score (blocks 4, 5, and 6). In this paper, we focus on the chords pre-classification, and assume that the Sound Segmentation is already available.

A. Chords Recognition Module

The Chords Recognition Module (block 3, figure 3) is detailed in figure 4.

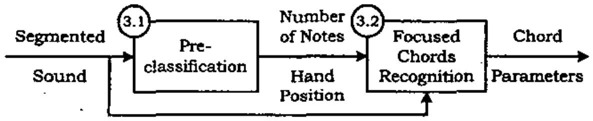

Fig. 4. Block diagram of the Chords Recognition Module

Here, the chords recognition task is divided into two processing modules, which are the "Pre-classification" and the "Focused Chords Recognition" (figure 4). The Pre-classification Module aims at reducing the burden of the Recognition Module. To achieve this, our solution is to reduce the complexity and increase the robustness of the recognition algorithm by using a smaller search domain.

IV. Proposed Pre-classification System

In designing the pre-classification algorithm, we aimed at obtaining two parameters from an observed audio segment : 1) the hand position on the guitar neck ; and 2) the number of notes in the chord. Finding a way to detect how many notes are present in a chord is usually a great challenge. An appropriate domain to perform this discrimination is the energy distribution by critical bands of the signal. The differences between critical bands energy bar charts are visible from one hand position to another. The energy distribution also varies depending on the number of notes in the chord. Based on this observation, we propose to use six parallel neural networks with energy vectors as an input. Each network is trained for a specific number of notes. Figure 5 shows the proposed pre-classification system. The output of the neural networks and their processing are explained in section B.

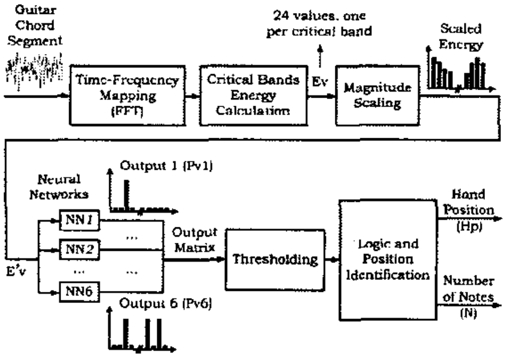

Fig. 5. Block Diagram of the Proposed Pre-classification System

The system input is a frame of audio samples. In order to have a sufficient frequency resolution, the frame length chosen is 2048 samples [11]. At a sampling frequency of 44.1 kHz, this represents 46.4 milliseconds. The output of the system is : 1) the hand position (Hp) on the guitar neck ; and 2) the number of notes (N) in the chord.

B. Blocks Description

The proposed Pre-classification System is detailed in the following sections. Each module is described to follow the signal and data processing along the block diagram (fig. 5).

Time-Frequency Mapping

A single Fast Fourier Transform is used to obtain the spectrum amplitude of the signal segment. The output vector is then used for energy calculation.

Critical Bands Energy Calculation

The energy calculation follows a psychoacoustic critical bands distribution, the Barkhausen Distribution [10], where units are the "Barks" (figure 6).

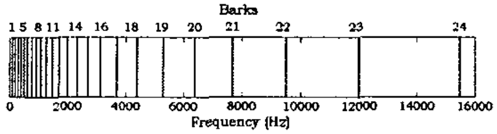

Fig. 6. Barkhausen Critical Bands Frequency Distribution

The spectrum vector of the signal segment is used to compute a 24 dimensional, real-valued energy vector Ev (figure 5), where each entry is the energy of the corresponding critical band described in table 1.

TABLE 1
Critical Bands Energy Distribution (Hertz)

#*	[B,E[**	#	[B,E[	#	[B,E[
1	[0,100[	9	[920,1 080[	17	[3 150,3 700[
2	[100,200[	10	[1 080,1 270[	18	[3 700,4 400[
3	[200,300[	11	[1 270,1 480[	19	[4 400,5 300[
4	[300,400[	12	[1 480,1 720[	20	[5 300,6 400[
5	[400,510[	13	[1 720,2 000[	21	[6 400,7 700[
6	[510,630[	14	[2 000,2 320[	22	[7 700,9 500[
7	[630,770[	15	[2 320,2 700[	23	[9 500,12 000[
8	[770,920[	16	[2 700,3 150[	24	[12 000,15 5000[

*Band number, **Begin, End

The Ev vector covers a 15.5 kilohertz bandwidth, corresponding to the cochlear frequency distribution.

Magnitude Scaling

The "Magnitude Scaling" block transforms the energy vector on a dB scale to enhance its high-frequency components. The neural network used needs inputs to be in]0,1[. The output energy vector E'v (figure 5) is thus normalized between minimum and maximum values, 0.1 and 0.7. These values are determined empirically to ensure training algorithm convergence.

Neural Network System Core

The Neural Network Core is based on feed-forward perceptron topology, which is detailed as follows :

- Two hidden layers
- Log-sigmoid transfer functions
- Input layer : 24 cells
- First, second hidden layers : 31, 24 cells
- Output layer : 19 cells (19 possible hand positions)

The input of the neural networks is the 24-dimensional scaled energy vector E'v (figure 5), from which each of the six neural networks calculates an output vector. The output vector Pv (figure 5), a nineteen-dimensional, real-valued vector is, once rounded to the nearest integer, formed of nineteen binary values (ones and zeros). These nineteen values correspond to the possible hand positions on the guitar neck for a fixed typical hand width of five frets (maximum for most chords). Figure 7 shows two examples of hand positions for the five-frets hand width.

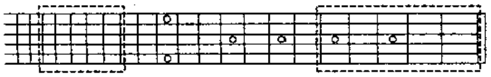

Fig. 7. Hand position on the guitar neck

The number of possible hand positions is explained as follows. There are 22 frets on the fingerboard. We add to this the "0" position, which means open strings. Then, assuming a maximum hand width of 5 frets, there are 23 – 5 + 1 = 19 possible hand positions.

Logic and Position Identification (algorithm decision)

Once the outputs of the 6 neural networks are calculated (Pv1 to Pv6 in figure 5), the following logic is applied :

```
1. $S_m$ = sum(Pv(m)), M = min($S_m$), and N = $\underset{m}{\text{argmin}}(S_m)$

2. if M = 1                                              (CASE 0)
      Hp = position of non-zero entry in Pv(N)
   end
   if M > 1                                              (CASE 1)
      Pa = lowest index of non-zero entry in Pv(N)
      Pb = highest index of non-zero entry in Pv(N)
      R = Pb – Pa = range of focused search
   end
   if $S_m$ has more than one values                     (CASE 2)
   at the lowest level
      ⇒ ambiguity (no decision)
```

Fig. 8. Logic and position Identification Algorithm

Ideally, only one out of the six neural networks should have an output with only one non-zero entry, whose index describes the hand position on the guitar neck. This algorithm first determines how many notes are forming the analyzed chord (N, figure 5), by summing (S_m) the elements of the output position vector Pv (figure 5). The lowest level S_m determines the number of notes. The hand position (Hp) corresponds to the position of the non-zero entry in Pv(N). If more than one possible positions are found for the lowest level S_m (M > 1), the possible Hp are treated between Pa and Pb, over a range R described as [Pa,Pb].

C. Neural Network Core Supervised Training

The neural networks used are trained to classify the chords by hand position. Each neural network is specialized for a given number of notes. To train the neural networks, we use a database formed by random synthetic chords sequences. The objective is to avoid overtraining of the networks. Because of the specialization of each of the six neural networks, the number of notes forming the analyzed musical chord is implicitly obtained from the neural network number (NN1 to NN6 in figure 5).

The neural networks are trained with known input/output pairs (supervised learning [2]). A specialized database is used for each neural network, as follows :

- Neural Network #1 (NN 1) : one string (note) only
- NN 2 : Exactly two notes

...

- NN 6 : Exactly six notes

The training database elements are synthetic signals modeled after real signals as per section II. Thirty synthetic chords are used to train the neural networks for each hand position and number of notes. The training dataset is then composed of 19 x 6 x 30 = 3420 synthetic chords.

The training is, obviously, an important phase of this project. The training dataset, the performance function and the performance criterion of this function are critical characteristics to set. The following sections will illustrate how the neural networks system was designed.

Training Algorithm

The training algorithm phase uses is the Scaled Conjugate Gradient algorithm (SCG), because of its well known performance in classification tasks. In the conjugate gradient algorithms, a search is performed along conjugate directions, which produces generally faster convergence than a search along the steepest descent directions. The conjugate gradient algorithms, in particular the SCG, perform well over a wide variety of problems, particularly for networks with a large number of weights. Also, the SCG algorithm is almost as fast as the Levenberg-Marquardt (LM) algorithm on function approximation and classification problems [2,8].

Training Dataset

In order to prepare the training dataset, various criteria have to be considered. On the guitar neck, the first string is the thinnest (high-pitched notes); while the sixth string is the biggest (low-pitched notes). The synthetic chords generation is formed by the following steps :

- Select one hand position and the number of notes
- Randomly choose fingering charts with the constraint of a fixed number of notes (1 to 6), depending on the neural network under training
- Generate the corresponding chords from sinusoid signals, by adding up to twelve harmonic signals with the fundamental frequency signal of each note
- Convolve with spectral envelope (figure 1)

Training Performance Function

Each network is trained with the standard mean squared error (*mse*) performance function. Equation 1 describes the *mse*, where e_i is the error between the target t_i and the neural network output a_i. The training stops when the desired *mse* is reached. The desired *mse* is an empirically determined threshold making the training converge in a reasonable time.

$$mse = \frac{1}{N}\sum_{i=1}^{N}(e_i)^2 = \frac{1}{N}\sum_{i=1}^{N}(t_i - a_i)^2 \qquad (1)$$

We use a 1/275 *mse*, in order to obtain good performance with about 7000 epochs per training.

V. Classifier Performance

Table 2 summarizes the results. Hp is the known hand position. For example, when Hp = 1, line 1 of table 2 indicates that the pre-classification algorithm correctly estimates the hand position for 96.9% of the chords with N = 1 note, 96.8% of the chords with N = 2 notes, etc.

TABLE 2
Classifier Performance (%) by Hand Position

	Number of notes					
Hp*	1	2	3	4	5	6
1	96.9	96.8	99.0	98.2	95.8	96.0
2	96.7	96.1	98.9	98.0	95.8	95.7
3	96.3	96.1	98.9	97.9	95.7	95.7
4	95.8	96.0	98.9	97.8	95.6	95.7
5	94.6	95.4	98.8	97.6	95.5	95.6
6	94.5	95.3	98.7	97.5	95.2	95.6
7	94.0	95.1	98.6	97.2	95.0	95.4
8	93.3	94.5	98.2	96.9	94.8	95.3
9	92.8	94.4	98.1	96.6	94.4	95.1
10	92.4	94.3	97.9	96.6	94.4	94.8
11	90.3	94.0	97.7	96.5	94.3	94.8
12	90.3	94.0	97.7	96.3	94.3	94.7
13	90.2	93.4	97.5	95.9	94.2	94.6
14	90.0	93.1	97.1	95.9	93.8	94.4
15	89.9	92.0	95.8	95.8	93.7	94.4
16	89.8	91.6	95.4	95.7	93.6	93.9
17	89.2	90.4	95.3	95.6	93.6	93.7
18	89.0	90.1	95.2	95.1	93.5	93.6
19	89.0	89.9	95.0	95.1	93.5	93.1

*Hand Position

For each hand position (19 in all), and for each number of notes (1 to 6), two thousand randomly generated chords were used in the test set. An average performance of 94.96% accuracy for synthetic chords has been measured. The remaining 5.04% fell in case 1 and case 2 (figure 8). The performance is high enough to think about real chords pre-classification. Figure 9 shows the classifier performance for each number of notes in the tested chords.

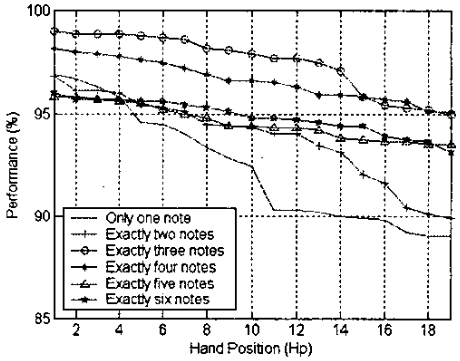

Fig. 9. Classifier Performance (%) by Hand Position

VI. Conclusion

We presented a neural network based classification approach to reduce the task of chords recognition in a music transcription system. The neural network pre-classifier works properly on synthetic guitar chords, with some limitations. Ambiguities have to be analyzed to improve the actual system performance.

References

[1] Bonnet, L., "High-Resolution Robust Multipitch Analysis of Guitar Chords," *114th Convention of the Audio Engineering Society*, Convention Paper 5772, Amsterdam, The Netherlands, 2003.

[2] Haylin, S., *Neural Networks, A Comprehensive Foundation, Second Edition*, Upper Saddle River: Prentice Hall, 1999.

[3] Kaiser, J. F., "Detection of transient signals using the energy operator," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3, pp. 145-148, Minneapolis, 2001.

[4] Kashino, K., Tanaka, H., "A sound source separation system with the ability of automatic tone modeling," *Proceedings of the International Computer Music Conference*, pp. 248-255, 1993.

[5] Klapuri, A.P., Eronen, A., Sapparanen, J, Virtanen, T., "Automatic Transcription of Music," *Proceedings of the 14th Meeting of the FWO Research Society on Foundations of Music Research*, Ghent, Belgium, Oct. 2001.

[6] Klapuri, A, Virtanen, T., "Separation of Harmonic Sound Sources Using Sinusoidal Modeling," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 765-768, Istanbul, 2000.

[7] Martin, K. D., "Automatic transcription of simple polyphonic music : Robust front-end processing," *MIT Media Laboratory, Perceptual Computing Section. Technical Report*, pp. MA : 399, 1996.

[8] MATLAB, "Neural Networks User's Guide," The Mathworks inc., 2002.

[9] Moorer, J. A., "On the segmentation and analysis of continuous musical sound by digital computer," *Center of Computer Research in Music and Acoustics, Department of Music, Stanford University*, no. Report no. STAN-M-3, 165 p., May 1975.

[10] Smith, J. O., Abel, J. S., "The Bark and ERB Bilinear Transforms," *Preprint of version accepted for publication in the IEEE Transactions one Speech and Audio Processing*, Dec. 1999.

[11] Tolonen, T., "A computationally Efficient Multipitch Analysis Model," *Proceedings of the IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 708-716, Nov. 2000.